Machine Learning Engineer Nanodegree

Capstone Proposal: Bertelsmann/Arvato Project

Mohammed Tayor
13 February 2020

Project Overview

This project domain is customer identification and segmentation through marketing and demographic data. The data about marketing campaigns and the general population of the German country demographics is provided by Arvato Financial Solutions[1].

Problem Statement

This project has two main goals. In the first phase, we are going to do a customer segmentation[2] using unsupervised learning of the German population based on demographic data provided by Bertelsmann Arvato Analytics. In a second phase, we are going to use supervised learning techniques namely classification task algorithms to determine the population who is likely to convert into customers from a marketing email of the same company.

Datasets and Inputs

The dataset provided consists of four CSV files. All of the files contain demographic data about the German population. The largest one (Udacity_AZDIAS_052018.csv) was made as a general overview of the whole German population (891 211 persons x 366 features). The second (Udacity_CUSTOMERS_052018.csv) represents a portion that is the customer base of a mail-order company (191 652 persons x 369 features). Both will be used for the first task of customer segmentation. Then we have a file (Udacity_MAILOUT_052018_TRAIN.csv) carrying demographics of individuals who were targets of a marketing campaign with binary results (42 982 persons x 367 columns), and a last one (Udacity_MAILOUT_052018_TEST.csv) representing people who were targets of a marketing campaign without providing the results for which we are asked to develop a supervised model to predict the outcome.

Solution Statement

As we are dealing with a relatively big dataset with a lot of features and some missing values, we need to preprocess and do some kind of dimensionality reduction using techniques like PCA[3], t-SNE[4] or UMAP[5]. Then we can do our clustering using K-Means[6] as it is the most common algorithm for such tasks of customer segmentation. For the second part of supervised learning, we will try to ensemble different kinds of models to make our predictions better, namely, XGBoost[7], LightGBM[8], Sklearn and we will see if we have good results using deep learning Keras[9] framework too. The models will need also to be optimized so we will use Bayesian optimization library Hyperopt[10] for hyperparameter optimization.

Benchmark Model

The unsupervised model should give a good result by comparing it with how likely it is going to identify the existing population described in the second CSV file as one cluster of the first CSV (the German population).

For the second model, we are going to benchmark it with the results of the public leaderboard of the Kaggle competition that is running. At the time of the writing of this proposal, the best achieved AUR score is 0.81063. We will also see how other automated machine learning tools compare to our solutions, like Auto-sklearn[11] python package.

Evaluation Metrics

The project metric will be the AUR [12] (Area Under the Curve) for the ROC curve (Receiver operating characteristic) as described by the Kaggle competition evaluation page.

Project Design

The first and most fundamental step to deal with this project would be to do an exploratory data analysis, see what values are missing if there is an imbalance of the dataset between positive and negative targets, or any outliers detected. This step would mainly be done using Pandas. Then we are going to get more insights into the data using visualizations of the Seaborn[13] package. The second step would be to get the data into a more convenient shape for the next phases. For that, we need to preprocess it and solve any issues found in the last step like removing outliers, reducing the feature space, and downsampling or upsampling the imbalanced data using a package like Imbalanced-learn[14]. The third step would be to do the clustering or the customer segmentation utilizing K-Means. The fourth step consists of developing our classification model using the ensembling technique[15] of XGBoost, LightGBM and Sklearn library (random forest for example) and do the proper hyperparameter optimization of them with proper cross-validation strategy and compare it to the automated modeling solutions like Auto-sklearn. Finally, we submit our solution to the Kaggle competition for a benchmark of how we perform compared to the best solution so far.

References

[1] “Arvato - Bertelsmann SE & Co. KGaA.” [Online]. Available: https://www.bertelsmann.com/divisions/arvato/. [Accessed: 13-Feb-2020].

[2] “Market segmentation,” *Wikipedia*. 24-Dec-2019.

[3] "sklearn.decomposition.PCA — scikit-learn 0.22.1 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html. [Accessed: 13-Feb-2020].

[4] "sklearn.manifold.TSNE — scikit-learn 0.22.1 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html. [Accessed: 13-Feb-2020].

[5] "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction — umap 0.3 documentation." [Online]. Available: https://umap-learn.readthedocs.io/en/latest/. [Accessed: 13-Feb-2020].

[6] "sklearn.cluster.KMeans — scikit-learn 0.22.1 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. [Accessed: 13-Feb-2020].

[7] "XGBoost Documentation — xgboost 1.0.0-SNAPSHOT documentation." [Online]. Available: https://xgboost.readthedocs.io/en/latest/. [Accessed: 13-Feb-2020].

[8] "Welcome to LightGBM's documentation! — LightGBM 2.3.2 documentation." [Online]. Available: https://lightgbm.readthedocs.io/en/latest/. [Accessed: 13-Feb-2020].

[9] "Home - Keras Documentation." [Online]. Available: https://keras.io/. [Accessed: 13-Feb-2020].

[10] *hyperopt/hyperopt*. hyperopt, 2020.

[11] "auto-sklearn — AutoSklearn 0.6.0 documentation." [Online]. Available: https://automl.github.io/auto-sklearn/master/. [Accessed: 13-Feb-2020].

[12] "Receiver operating characteristic - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic#Area_under_the_curve. [Accessed: 13-Feb-2020].

[13] "seaborn: statistical data visualization — seaborn 0.10.0 documentation." [Online]. Available: https://seaborn.pydata.org/. [Accessed: 13-Feb-2020].

[14] "Welcome to imbalanced-learn documentation! — imbalanced-learn 0.5.0 documentation." [Online]. Available: https://imbalanced-learn.readthedocs.io/en/stable/. [Accessed: 13-Feb-2020].

[15] "Kaggle Ensembling Guide | MLWave." [Online]. Available: https://mlwave.com/kaggle-ensembling-guide/. [Accessed: 13-Feb-2020].